Theoretical Computer Science 28 (1984) 1–12
North-Holland

ON HEADS VERSUS TAPES

W. PAUL*
IBM Research Laboratory, San Jose, CA 95193, U.S.A.

Communicated by A. Meyer
Received 15 May 1981
Revised 3 May 1982

Abstract. For d and $k \geq 2$, d-dimensional k-tape Turing machines cannot simulate d-dimensional Turing machines with k heads on 1 tape in real time.

1. Introduction

In this paper we argue that replacing k storage units each with one access point by 1 storage unit with k access points increases the power of the computing machine, to which these units are attached. Formally, our computing machine will be a Turing machine, our storage units will be multi-dimensional tapes and access points will be heads. They can be viewed as models for array-like data structures or storage units, where in t steps $\omega(t)$ items of data are readable. Machines with k heads on one d-dimensional tape will be called *d-dimensional k-head machines.*

Let M be a machine that, given sequences $E = e_1e_2 \cdots$ of input symbols on an input tape, produces sequences $A = a_1a_2 \cdots$ of output symbols on an output tape. M works *on-line* if for all inputs E the computation of M given E proceeds in stages $1, 2, \ldots$ such that for all i during the ith stage input symbol e_i is read, some computation is performed and output symbol a_i is printed. M works in *real time* if there is a constant ξ such that, for all inputs E and all i, the ith stage of the computation of M given E consists of at most ξ steps. Machine S *simulates* machine M if for all inputs E machine S given E produces the same output as machine M given E.

In [5] a technique has been proposed that allows to study the effect of various storage structures on the computing power of T[illegible]g machines that work on-line. In particular it has been shown for all k and $d \leq 2$ [illegible] on-line simulation of machines with k d-dimensional tapes by machines with $(k-1)$ d-dimensional tapes requires nonlinear time. Here we will use techniques from [5] in order to give a complete proof of the following theorem (see Section 8).

* Part of this research was done while the author was working at the University of Bielefeld, Fed. Rep. Germany.

0304-3975/84/$3.00 © 1984, Elsevier Science Publishers B.V. (North-Holland)

Theorem 1. *2-dimensional 2-tape machines cannot simulate 2-dimensional 2-head machines in real time.*

A straightforward extension of the argument gives the following theorem.

Theorem 2. *For all k and $d \geqslant 2$, d-dimensional k-tape machines with any number of additional linear tapes cannot simulate d-dimensional k-head machines in real time.*

We compare this with known simulation results. One-dimensional k-head Turing machines can be simulated in linear time and on-line by 1-dimensional k-tape Turing machines [7]. They can also be simulated in real time by 1-dimensional multitape machines [2] and $(4k-4)$ tapes suffice for the simulation [4]. It is not known whether 1-dimensional k-head machines can be simulated in real time by 1-dimensional k-tape machines. Two-dimensional k-head machines can be simulated in linear time and on-line by 2-dimensional k-tape machines [6]. Together with Theorem 1.2 this shows that 2-dimensional k-tape machines that work on-line and in linear time are strictly more powerful than 2-dimensional k-tape machines, that work in real time. For $d > 1$, d-dimensional 2-head machines can be simulated in real time by Turing machines with 3 d-dimensional and some 1-dimensional tapes. For $d > 1$ and $k > 2$, d-dimensional k-head Turing machines can be simulated in real time by Turing machines with $\frac{3}{2}k(k-1)(k-2)$ d-dimensional and $O(k^3 d)$ 1-dimensional tapes [4]. Thus, for $k = 2$, Theorem 2 cannot be improved.

2. Descriptional complexity

We begin by developing a simple theory of descriptional complexity of rectangular figures, that is already implicit in [5].

(2.1) Self-delimiting strings. Let s be a binary string of length l and let $\mathrm{bin}(l)$ be the binary representation of l. Form $\overline{\mathrm{bin}(l)}$ by replacing in $\mathrm{bin}(l)$ each 0 by 00 and each 1 by 11. We call the string $s' := \overline{\mathrm{bin}(l)}01s$ the *self-delimiting version* of s.

Fact 1. *Let $s_1, \ldots, s_k$ be binary strings with lengths $l_1, \ldots, l_k$. Then $s'_1 \cdots s'_{k-1} s_k$ is a binary string of length $\sum_{i=1}^{k} l_i + O(\sum_{i=1}^{k-1} \log l_i)$ which codes $s_1, \ldots, s_k$.*

(2.2) Figures. An $n \times m$ matrix F with entries in $\{0, 1, B\}$ is called an *$n \times m$ figure.* $|F| := nm$ is called the *area* of F and (n, m) the *shape* of F. Figures correspond in an obvious way to rectangular parts of inscriptions of 2-dimensional Turing machine tapes with alphabet $\{0, 1, B\}$, where B is the blank symbol.

A figure G is called a *subfigure of F at position (a, b)* if $G_{i,j} = F_{i+a,j+b}$ for all i and j. For $i = 1, 2$ let F^i be $n_i \times m$ figures. We define (F^1, F^2) to be the $(n_1 + n_2 + 1) \times$

$\max\{m_1, m_2\}$ figure specified by Fig. 1. For figures $F^1, \ldots, F^k$ we define

$$(F^1, \ldots, F^k) = (\cdots((F^1, F^2), F^3), \ldots, F^k).$$

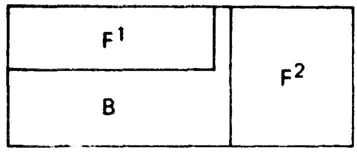

Fig. 1.

We fix a simple ordering among figures. Let $<_{\text{lex}}$ be the lexicographic ordering based on $B < 0 < 1$. For $i = 1, 2$ let F^i be an $n_i \times m_i$ figure and

$$s^i = F^i_{11} \cdots F^i_{1n_i} \cdots F^i_{m_i1} \cdots F^i_{m_in_i} \in \{0, 1, B\}^{n_im_i}$$

the string which is obtained by concatenating the rows of F^i. We define $F^1 < F^2$ if $(n_1, m_1) <_{\text{lex}} (n_2, m_2)$ or $(n_1, m_1) = (n_2, m_2)$ and $s^1 <_{\text{lex}} s^2$. Occasionally we will treat sets D of figures like sequences of figures. In such cases the sequence of figures in D ordered by $<_{\text{lex}}$ is meant.

(2.3) Kolmogorov complexity of figures. Let C be the class of Turing machines with one 1-dimensional input tape, one 2-dimensional working tape and tape alphabet $\{0, 1, B\}$ on both tapes. For $M \in C$ let $c(M)$ denote the self delimiting version of the standard encoding of M into $\{0, 1\}^*$. Let U be a universal machine in C, i.e., for any $M \in C$ and $v \in \{0, 1\}^*$ the machine U started with input $c(M)v$ will simulate M with input v. Let $F_1, \ldots, F_s, G_1, \ldots, G_t$ be figures. The *Kolmogorov complexity* $K(F_1, \ldots, F_s | G_1, \ldots, G_t)$ *of* $F_1, \ldots, F_s$ *given* $G_1, \ldots, G_t$ is defined as the length of the shortest $x \in \{0, 1\}^*$ such that U started with x on the input tape and $(G_1, \ldots, G_t)$ on the working tape with the head on the top left corner of G_1 produces $(F_1, \ldots, F_s)$ on the working tape and halts.

The *Kolmogorov complexity* $K(F_1, \ldots, F_s)$ of $(F_1, \ldots, F_s)$ is defined as $K(F_1, \ldots, F_s | \lambda)$ where λ is the empty figure, i.e., the unique figure whose shape is $(0, 0)$. Intuitively, $K(F_1, \ldots, F_s | G_1, \ldots, G_t)$ is the number of bits necessary to specify $F_1, \ldots, F_s$ if $G_1, \ldots, G_t$ are known. Also

$$I(F_1, \ldots, F_s | G_1, \ldots, G_t) := K(F_1, \ldots, F_s) - K(F_1, \ldots, F_s | G_1, \ldots, G_t)$$

is intuitively the number of bits which are saved by the knowledge of $G_1, \ldots, G_t$ if we wish to specify $F_1, \ldots, F_s$. Consequently it is called the *information about* $F_1, \ldots, F_s$ *in* $G_1, \ldots, G_t$. We will not use it here formally, but in order to support intuition we will occasionally rephrase and interpret intermediate results involving Kolmogorov complexity in terms of information. Some immediate consequences of the definitions and Fact 1 are summarized in the following.

Fact 2. *Let F, G and H be any figures. Then*

$$K(F\mid H)\leqslant K(F,G\mid H)+\mathrm{O}(\log|F|),$$

$$K(G\mid H)\leqslant K(F,G\mid H)+\mathrm{O}(\log|G|),$$

$$K(F\mid G,H)\leqslant K(F\mid G)+\mathrm{O}(\log|G|),$$

$$K(F\mid H)\leqslant K(F\mid G)+K(G\mid H)+\mathrm{O}(\log|F|),$$

$$K(F,G\mid H)\leqslant K(F\mid H)+K(G\mid H)+\mathrm{O}(\log|F|).$$

Fact 3. *Let F^1 and F^2 be figures of the same shape which differ in at most k entries. Let G be a third figure. Then*

$$K(F^2\mid F^1)\leqslant \mathrm{O}(k\log|F^1|),\qquad |K(G\mid F^1)\text{-}K(G\mid F^2)|\leqslant \mathrm{O}(k\log|F^1|).$$

A simple but important consequence of Fact 3 on Turing machine computations is the following.

Fact 4. *Let S be a 2-dimensional multitape Turing machine with tape alphabet $\{0,1,B\}$. Let s be a natural number and for $i\in 1,\ldots,s$ let C_i be the inscription of a rectangular portion of some tape of S. Suppose in some time interval for all i the portion of tape occupied by C_i was* **visited** *at most k_i times, i.e., during at most k_i steps during that interval there was a head on some cell of C_i, and let C_i' be the resulting inscription. Then*

$$K(C_1',\ldots,C_s'\mid C_1,\ldots,C_s)\leqslant \mathrm{O}(\textstyle\sum k_i\log|C_i|).$$

Consequently if F is any figure, then

$$|K(F\mid C_1,\ldots,C_s)-K(F\mid C_1',\ldots,C_s')|\leqslant \mathrm{O}(\textstyle\sum k_i\log|C_i|).$$

Phrased in terms of information this says that, by visiting the portion of tape occupied by $C_1,\ldots,C_s$ $\sum k_i$ times, one can pump no more than $\mathrm{O}(\sum k_i\log|C_i|)$ bits of information about figure F into that portion. Also observe that we have here a function from configurations into the natural numbers whose value changes only little in a sigle computation step.

(2.4) Random squares. An $n\times n$-0/1 matrix Q is called an $n\times n$-0/1 *square*. It is called *(Chaitin-)random* if $K(Q)\geqslant n^2$. Since there are 2^{n^2} $n\times n$-0/1 matrices and at most $2^{n^2}-1$ binary strings of length less than n^2, random squares of all sizes exist. Basically random squares are their own shortest descriptions. For lower bound proofs they have the very desiraible property that all coding tricks for such squares are more or less obvious or impossible. We make this now somewhat more precise.

Fact 5. *Let Q be an $n\times n$–0/1 random square and let $P_1,\dots,P_s$ be pairwise nonoverlapping subfigures of Q. Let Q' be obtained from Q by replacing for each i an occurrence of P_i in Q by blanks. Then*

$$K(P_1,\dots,P_s\mid Q')\geq \sum |P_i| - O(s\log n).$$

Proof. Q can be specified by n, the shapes and positions of $P_1,\dots,P_s$ in Q, the bits of Q' in row order and how to produce $P_1,\dots,P_s$ if Q' is known. Thus

$$n^2\leq K(Q)\leq O(s\log n)+n^2-\sum |P_i| + K(P_1,\dots,P_s\mid Q'). \qquad \square$$

Phrased in terms of information this says that a reasonably regularly shaped portion Q' of a random square Q contains very little information about the missing portions $P_1,\dots,P_s$ of Q. Taking $s=1$ and applying Fact 1 we get $K(P_1)\geq |P_1|-O(\log n)$. Thus for sufficiently big neighborhoods random squares are locally almost random.

Fact 6. *Let Q be an $n\times n$–0/1 random square and let $P_1,\dots,P_s$ be pairwise nonoverlapping $p\times p$ subsquares of Q. Let R be any figure. Then*

$$\max K(P_i\mid R)\geq p^2 - O(|R|/s) - O(\log n).$$

Proof. Q can be specified by n, p, the positions of $P_1,\dots,P_s$ in Q, R, for each i how to produce P_i if R is known and finally the remaining bits of Q in row order. Thus

$$n^2\leq K(Q)\leq O(s\log n)+O(|R|)+\sum K(P_i\mid R)+n^2-sp^2. \qquad \square$$

This says that a figure R can obtain much information only about a constant times as many subsquares P_i of Q as its area can cover.

3. Definition of a machine M with 2 heads on one tape and a basic property of its simulators

M has linear input and output tapes and one 2-dimensional working tape. The working and the output alphabets are $\{0, 1, B\}$. An *action* of M is to move one of its heads (we allow diagonal moves) or to print a symbol under one of its heads or to output the symbol under one of its heads on the working tape. We associate an input symbol with each action. Upon reading an input symbol M performs the corresponding action. M is an abstract storage unit in the sense of [5].

Let S, the 'simulator', be a 2-dimensional 2-tape Turing machine with working alphabet $\{0, 1, B\}$ and suppose S simulates M in real time with delay ξ, i.e., S makes at most ξ steps to simulate any step of M.

A *block* is a set of tape cells of M or S that forms a square. At any given time t in a computation the content of a block B is a figure, which we denote by 'B at time t'. When it is clear which time t is meant, we will use the same notation B for both block B and its content at time t.

Fact 7. *Let l be any integer and J any input sequence for M and S. Consider the configuration of M and S after each has processed J. In this configuration of M, for $i = 1, 2$, let the figure B_i be the content of a block reachable by head i of M within l steps. In this configuration of S, for $i = 1, 2$ let the figure C_i be the content of a block that contains all cells reachable by head i of M within ξl steps; then*

$$K(B_1, B_2 \mid C_1, C_2) \leq O(\log |(C_1, C_2|).$$

Proof. If (C_1, C_2) is known, (B_1, B_2) can be described in the following way:

(i) state and head positions of S relative to C_1 and C_2: $O(\log |(C_1, C_2|)$ bits,

(ii) for each input sequence J' of length l which drives a head of M somewhere into B_1 or B_2 and then prints out the symbol under that head run S on that input sequence. One has to use only $O(\log l) \leq O(\log |(C_1, C_2)|)$ bits in order to specify the relative positions of B_1 and B_2 to the heads of M plus an additional $O(1)$ bits for a little simulation program which generates from the given data all the sequences J' and runs S on input J'. $\square$

In terms of information this fact states that if there is an information deficit about (B_1, B_2) in (C_1, C_2), then one cannot answer all questions concerning (B_1, B_2) by looking at (C_1, C_2) only.

4. The input sequence for M

We specify an input sequence I for M whose last steps cannot be simulated with delay ξ and outline the proof of Theorem 1. Sequence I will consist of 5 parts $I_1 \cdots I_5$ where I_2, I_4 and I_5 will depend on the behavior of the simulator.

(4.1) Choose a large $n \times n$–0/1 random square Q. Partition it into $n^{3/4} \times n^{3/4}$ *blocks*. Let R be the figure which is obtained by replacing the bottom right block L of Q by blanks. Part I_1 makes head 1 of M print R row by row and then moves head 1 to the top left corner of L. Head 2 stays at the top left corner of R.

(4.2) Let $t = n^{7/8} \log^2 n$ and let T be the top left $t \times t$ subsquare of Q (resp. R). It consists of $n^{1/4} \log^4 n$ blocks. Let L_1 consist of the first $n^{1/2} \log^2 n$ rows of the bottom right block L of Q. Let L_2 consist of the next $n^{1/2}$ rows of L, L_3 of the next $n^{1/2}$ rows and so on. L_i is called *layer i*. Thus, layer 1 is larger than the other layers by a factor $\log^2 n$ and there are $\tau \approx n^{1/4}$ layers.

Part I_2 consists of τ parts $J_1, \ldots, J_\tau$ where each part J_i consists of 2 parts $M_i W_i$. The 'moving part' M_i has length at most t and drives head 2 of M to the top left corner of some block B_i in T. The choice of B_i will depend on the behavior of S during $I_1 J_1 \cdots J_{i-1}$ (M_1 is empty). In the 'writing part' W_i, head 1 of M writes down L_i row by row (see Fig. 2). Observe

$$|M_1 \cdots M_\tau| = O(n^{9/8} \log^2 n).$$

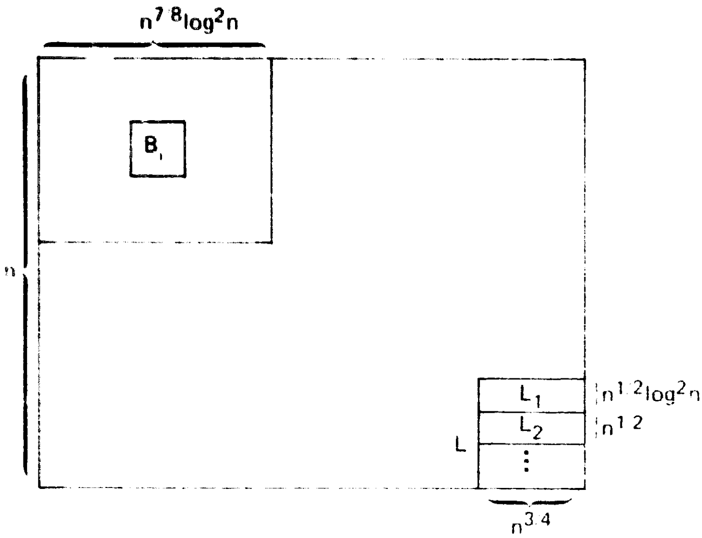

Fig. 2.

The effect of M_i will be that after execution of M_i at least one head of S is far away from those portions of the tape of S that contain much information about L (see Lemma 1 of Section 6). Because L_1 is much bigger than the other layers we will be able to show that it is the same head for all i, say head 2 of S, that is far away from information about L after M_i (Lemma 4). Thus, while head 1 of S may be able to store information about L in a compact way, head 2 must spread it all over its tape (Lemma 5).

(4.3) Part I_3 drives both heads of M to the top left corner of L. Now we make use of the fact that both heads of M can read what only one head of M did write, namely L.

(4.4) Partition L into small $n^{11/16} \times n^{11/16}$ blocks. Depending on the behavior of S in $I_1 I_2 I_3$ a pair (a, b) of the small blocks is chosen. Because information about L is spread out over tape 2 of S, it will be possible to choose (a, b) such that in all small neighborhoods of S there will be information deficit about a or information deficit about b.

Part I_4 drives the heads of M to the top left corners of a and b.

(4.5) As a consequence of Fact 7 there will be at least one choice for I_5 of $n^{11/16}$ moves which cannot be correctly simulated by S with $\xi n^{11/16}$ further steps.

5. Choosing the blocks B_i

Partition the tapes of S into $\xi n^{3/4} \times \xi n^{3/4}$ blocks. For sets D of blocks of S and natural numbers i we define $N_i(D)$ as the set of blocks reachable from D within $i\xi n^{3/4}$ moves. It is called the *i-neighborhood* of D. Clearly for fixed i we have $|N_i(D)| = O(|D|)$.

A block of S is called *fat* at time t if at that time it has been visited in at least $n^{5/4}/\log^2 n$ steps of I_2. We denote by fat(i) the set of blocks of S which is fat after execution of J_i. Now

$$|I_2| = \sum |M_i| + \sum |W_i| = O(n^{9/8} \log^2 n) + O(n^{3/2}) = O(n^{3/2}).$$

Thus at most $O(n^{3/2-5/4} \log^2 n) = O(n^{1/4} \log^2 n)$ blocks of S will ever be fat. Hence, for all i, the set of blocks $N_3(\text{fat}(i))$ ordered and interpreted as a figure has area at most $O(n^{3/2+1/4} \log^2 n) = O(n^{7/4} \log^2 n)$. For sets D of blocks of S and parts J of input sequences for M and S we denote by 'D after J' the set of contents of the locks in D in the configuration of S immediately after S has processed J. By Fact 6 we can choose a block B_{i+1} in T such that

$$K(B_{i+1} \mid N_3(\text{fat}(i)) \text{ after } J_i) \geq \tfrac{1}{2} n^{3/2}.$$

6. The effect of I_2 and I_3

The goal behind moving head 2 of M after J_i to a block B_{i+1} such that the vicinity of the fat blocks has large information deficit about B_{i+1} is of course to force at least one head of S far away from the fat blocks; recall that after M_{i+1} the simulator S must be ready to answer questions about B_{i+1} quickly. On the other hand head 1 of M stays close to L_1 and thus S must also be ready to answer questions about L_1 quickly. But at least intuitively only fat blocks can have enough information about L_1, thus we expect one head of S to be trapped in the close vicinity of at least one fat block. Precisely this is asserted in the first two lemmas.

Lemma 1. *For each $i \geq 1$ during all of W_{i+1} at least one head of S is outside of* $N_2(\text{fat}(i))$.

Proof. Suppose this assertion is false at time t during W_{i+1} and let C_1, C_2 be the blocks in $N_2(\text{fat}(i))$ visited by the heads of S at time t. Thus, $N_1(C_1, C_2) \subset N_3(\text{fat}(i))$.

By choice of B_{i+1} and Fact 2 we have

$$\begin{aligned}\tfrac{1}{2}n^{3/2} &\leq K(B_{i+1} \mid N_3(\mathrm{fat}(i)) \text{ after } J_i)\\ &\leq K(B_{i+1} \mid N_1(C_1, C_2) \text{ at time } t)\\ &\quad + K(N_1(C_1, C_2) \text{ at time } t \mid N_1(C_1, C_2) \text{ after } J_i)\\ &\quad + K(N_1(C_1, C_2) \text{ after } J_i \mid N_3(\mathrm{fat}(i)) \text{ after } J_i) + \mathrm{O}(\log n).\end{aligned}$$

By Fact 7 the first term on the right-hand side is $\mathrm{O}(\log n)$. By Fact 4 and because $|J_{i+1}| = \mathrm{O}(n^{5/4})$ the second term is $\mathrm{O}(n^{5/4} \log n)$. By Fact 2 the third term is $\mathrm{O}(\log n)$. □

Lemma 2. *For each $i \geq 1$ during all of J_{i+1} at least one head of S is in $N_1(\mathrm{fat}(i))$.*

Proof. Suppose this assertion is false at time t during J_{i+1}. Let C_1, C_2 be the blocks visited by S at time t. By Facts 5 and 2 we have

$$\begin{aligned}n^{5/4}\log^2 n - \mathrm{O}(\log n) &\leq K(L_1 \mid R)\\ &\leq K(L_1 \mid N_1(C_1, C_2) \text{ at time } t)\\ &\quad + K(N_1(C_1, C_2) \text{ at time } t \mid N_1(C_1, C_2) \text{ after } I_1)\\ &\quad + K(N_1(C_1, C_2) \text{ after } I_1 \mid R) + \mathrm{O}(\log n).\end{aligned}$$

By Fact 7 the first term on the right-hand side is $\mathrm{O}(\log n)$. No block in $N_1(C_1, C_2)$ is fat after J_i and $|J_{i+1}| = \mathrm{O}(n^{5/4})$, thus by Fact 4 the second term is $\mathrm{O}(n^{5/4} \log n)$.

Computing I_1 from R and simulating S on input I_1 gives all the tape inscriptions of S after I_1. $\mathrm{O}(\log n)$ further bits give $N_1(C_1, C_2)$ after I_1. Thus the third term is bounded by $\mathrm{O}(\log n)$. □

An immediate consequence is the following lemma.

Lemma 3. *For each $i \geq 1$ the head of S which is outside $N_2(\mathrm{fat}(i))$ after M_{i+1} does not enter $N_3(\mathrm{fat}(i))$ during W_{i+1}.*

Proof. Suppose it does enter $N_2(\mathrm{fat}(i))$ for the first time during W_{i+1} at time t. Then at time t it is still outside $N_1(\mathrm{fat}(i))$, thus, by Lemma 2, the other head is at time t inside of $N_1(\mathrm{fat}(i))$. But this contradicts Lemma 1. □

Now we know that for all i after M_{i+1} at least one head of S is outside $N_2(\mathrm{fat}(i))$ and stays there during all of W_{i+1}. W.l.o.g. let us assume it is head 2 of S for $i = 1$. The crucial point of the whole construction is that for all i it will be the same head.

Lemma 4. *For all $i \geq 1$ head 2 of S is outside $N_2(\mathrm{fat}(i))$ after M_{i+1}.*

Proof. The lemma is true for $i = 1$. Suppose it is true for all $i' < i$ and false for i.

Then after M_{i+1} by Lemma 1 head 1 of S is in some block $C_1 \notin N_2(\text{fat}(i))$ and by Lemma 2 head 2 of S is in some block $C_2 \in N_1(\text{fat}(i))$.

Let $k = \min\{k' \mid N_1(C_2) \cap \text{fat}(k') \neq \emptyset\}$ and let $E \in N_1(C_2) \cap \text{fat}(k)$. By induction hypothesis and Lemma 3 no block in $N_2(E)$ and hence no block in $N_1(C_2)$ was read (by head 2) during $W_{k+1}, \ldots, W_i$. Here we have used the crucial property that S has only one head on each tape. Intuitively the argument now is the following. Head 1 of S is not even close to a fat block. Thus it has for the next $\xi n^{3/4}$ steps access to only very little information about L_1 and L_2. The fat blocks which are accessible quickly by head 2 of S were not visited often before J_k, they were visited at most $|J_k|$ times during J_k and they were not visited after J_k except possibly in the relatively short periods $M_2, \ldots, M_{i+1}$. Thus if $k = 1$, then there was no opportunity to get enough information about L_2 into $N_1(C_1, C_2)$. But if $k > 1$, then only during J_k there was an opportunity to get $O(|J_k| \log n)$ bits about L_1 into $N_1(C_1, C_2)$. As $|L_1| = n^{5/4} \log^2 n$, this is not enough information.

The formal argument below follows exactly these lines.

If $k = 1$, then by Facts 5 and 2 we have

$$\begin{aligned} n^{5/4} - O(\log n) &\leq K(L_2 \mid Q \text{ without } L_2) \leq K(L_2 \mid R, I_1) + O(\log n) \\ &\leq K(L_2 \mid N_1(C_1, C_2) \text{ after } M_{i+1}) + K(N_1(C_1) \text{ after } M_{i+1} \mid R) \\ &\quad + K(N_1(C_2) \text{ after } M_{i+1} \mid N_1(C_2) \text{ after } J_1) \\ &\quad + K(N_1(C_2) \text{ after } J_1 \mid R, L_1) + O(\log n). \end{aligned}$$

By Fact 7 the first term on the right-hand side is $O(\log n)$. In $J_1 \cdots J_i M_{i+1}$ the blocks in $N_1(C_1)$ were visited at most $O(n^{5/4}/\log^2 n + n^{7/8} \log^2 n)$ times, thus by Fact 4 the second term is $O(n^{5/4}/\log n)$. In $J_2 \cdots J_i M_{i+1}$ the blocks in $N_1(C_2)$ can only have been visited in $M_2, \ldots, M_{i+1}$, thus by Fact 4 the third term is $O(n^{9/8} \log^3 n)$. The fourth term is clearly $O(\log n)$. This gives a contradiction in the case $k = 1$.

If $k > 1$, then by Facts 5 and 2 we have

$$\begin{aligned} n^{5/4} \log^2 n - O(\log n) &\leq K(L_1 \mid Q \text{ without } L_1) \\ &\leq K(L_1 \mid N_1(C_1, C_2) \text{ after } M_{i+1}) + K(N_1(C_1) \text{ after } M_{i+1} \mid R) \\ &\quad + K(N_1(C_2) \text{ after } M_{i+1} \mid R) + O(\log n). \end{aligned}$$

By Fact 7 the first term on the right-hand side is $O(\log n)$. Exactly as above, the second term can be estimated by $O(n^{5/4}/\log n)$. The blocks in $N_1(C_2)$ were visited at most $O(n^{5/4}/\log^2 n)$ times in $J_1 \cdots J_{k-1}$, at most $O(n^{5/4})$ times in J_k and at most $O(n^{9/8} \log^2 n)$ times in $M_{k+1} \cdots M_{i+1}$. Thus by Fact 4 the third term is $O(n^{5/4} \log n)$. This gives a contradiction in the case $k > 1$. $\square$

Lemmas 3 and 4 and $|I_i| = O(n)$ imply the following lemma.

Lemma 5. *In I_2I_3 no block of tape 2 of S is visited more than $O(n^{5/4}\log^2 n)$ times.*

7. The choice of I_4

We have forced the simulator S to spread out the information about L all over tape 2. Any block of tape 2 has very little information about L. For purposes of retrieving information about L tape 2 can intuitively be considered as degenerate and we are almost in the situation of 2 heads versus 1 head. Consequently, after an appropriate modification we will make the corresponding argument from [5] work.

Let C_1, C_2 be the blocks visited by the heads of S at the end of I_3. Partition $N_1(C_1, C_2)$ into small $\xi n^{11/16} \times \xi n^{11/16}$ blocks. Thus each of L, $N_1(C_1)$ and $N_1(C_2)$ is partitioned into $O((n^{3/4-11/16})^2) = O(n^{1/8})$ small blocks, each of area $n^{11/8}$ resp. $O(n^{11/8})$.

For sets D of small blocks of S let $\nu(D)$ be the set of blocks reachable from blocks in D within $\xi n^{11/16}$ steps. We say that a pair (c, d) of small blocks with c in $N_1(C_1)$ and d in $N_1(C_2)$ is *useful* for a pair (a, b) of small blocks in L if $K(a,b \mid \nu(c,d)) \le n^{11/8}/2$, i.e., if $\nu(c,d)$ contains a lot of information about (a, b). Our next goal is to find a pair (a, b) for which no pair (c, d) is useful. let $u(c, d)$ be the number of pairs (a, b) for which (c, d) is useful. Then we have the following lemma.

Lemma 6. *For all c, $\sum_d u(c,d) < n^{1/8}/\log n$.*

Proof. Let $\mu = n^{1/8}/\log n$ and suppose $\sum_d u(c,d) \ge \mu$ for some small block c in $N_1(C_1)$. Let $P_1, \ldots, P_\mu$ be distinct pairs of small blocks in L such that for each P_i there is a small block d_i in $N_1(C_2)$ such that (c, d_i) is useful for P_i. These pairs are formed of at least $\rho := \mu^{1/2} = n^{1/16}/(\log n)^{1/2}$ many distinct small blocks a_i of L. For each $j \le \rho$, let P_{i_j} be a pair where a_j occurs and let $e_j = d_{i_j}$. Then, by Fact 2,

$$K(a_j \mid \nu(c, e_j)) \le K(P_{i_j} \mid \nu(c, e_j)) + O(\log n) \le \tfrac{1}{2} n^{11/8} + O(\log n).$$

Now, by Facts 5 and 2 we have

$$\begin{aligned}
\rho(n^{11/8} - O(\log n)) &\le K(a_1, \ldots, a_\rho \mid Q \text{ without } a_1, \ldots, a_\rho) \\
&\le K(a_1, \ldots, a_\rho \mid R) + O(\rho \log n) \\
&\le K(a_1, \ldots, a_\rho \mid \nu(c), \nu(e_1), \ldots, \nu(e_\rho)) + K(\nu(c)) \\
&\quad + K(\nu(e_1), \ldots, \nu(e_\rho) \mid N_1(C_2) \text{ after } I_3) \\
&\quad + K(N_1(C_2) \text{ after } I_3 \mid R) + O(\rho \log n).
\end{aligned}$$

The first term on the right-hand side can be estimated by

$$\sum (K(a_j \mid \nu(c, e_j)) + O(\log n)) \le \rho(n^{11/8}/2 + O(\log n)).$$

The second term is $O(n^{11/8})$. The third term is $O(\rho \log n)$. By Lemma 5 and Fact 4 the fourth term is $O(n^{5/4} \log^3 n)$. □

By Lemma 6, $\sum_{c,d} u(c, d) = O(n^{1/8} n^{1/8}/\log n)$. Thus there is a pair (a, b) of small blocks of L for which no pair (c, d) of small blocks c in $N_1(C_1)$ and d in $N_1(C_2)$ is useful. Part I_4 of the input sequence is chosen to move the heads of M to the top left corners of a and b in $O(n^{3/4})$ steps.

8. Proof of Theorem 1

We are finally able to derive a contradiction from the assumption that S simulates M in real time.

Let (c, d) be the pair of small blocks visited by S after I_4. It was not useful for (a, b) after I_3. Hence by Fact 2 we have

$$\begin{aligned} n^{11/8}/2 &\le K(a, b \mid \nu(c, d) \text{ after } I_3) \\ &\le K(a, b) \mid \nu(c, d) \text{ after } I_4) + K(\nu(c, d) \text{ after } I_4 \mid \nu(c, d) \text{ after } I_3) \\ &\quad + O(\log n) \\ &\le O(\log n) + O(n^{3/4} \log n) + O(\log n) \end{aligned}$$

by Facts 7 and 4. □

References

[1] S.O. Aanderaa, On k-tape versus $(k-1)$-tape real time computation, in: R.M. Karp, ed., *Complexity of Computation* (Amer. Math Soc., 1974) pp. 75–96.

[2] M.J. Fischer, A.R. Meyer and A. Rosenberg, Real-time simulation of multihead tape units, *J. ACM* **19** (1972) 590–607.

[3] S.R. Kosaraju, Real-time simulation of concatenable double-ended queues by double-ended queues, *Proc. 11th Ann. ACM Symp. on Theory of Computing* (1979) pp. 346–351.

[4] B.L. Leong and J.I. Seiferas, New real-time simulations of multihead tape units, *J. ACM* **28** (1981) 166–180.

[5] W.J. Paul, J.I. Seiferas and J. Simon, An information theoretic approach to time bounds for on-line computation, *Proc. 12th ACM-STOC* (1980) pp. 357–367.

[6] W. Schnitzlein, Simulation von zweidimensionalen k-Kopf Turingmaschinen auf 2-dimensionalen k-Band Turingmaschinen, Preprint, Mathematics Dept., Universität Konstanz, 1981.

[7] H.J. Stoss, k-Band-Simulation von k-Kopf Turingmaschinen, *Comput.* **6** (1970) 309–317.